

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

FIG. 1 illustrates a flowchart illustrating a method according
5 to one embodiment of the invention;

FIG. 2 illustrates a computing system in which an embodiment of
the invention may be practiced;

10 FIG. 3 illustrates a block diagram illustrating a relationship
between update data and a row identifier index (RID);

FIG. 4A illustrates the result of a pre-associated to pending
insert operation at the conclusion of an update phase;

15 FIG. 4B illustrates the result of the pre-associated to pending
insert operation discussed in relation to FIG. 4A after a
commit phase;

20 FIG. 5 is a flow chart illustrating the update phase of an
insertion of a new element into a data structure according
to one embodiment of the invention;

FIG. 6 is a flow chart illustrating the commit phase of an insertion of a new element into a data structure according to one embodiment of the invention;

5 FIG. 7 illustrates a tree data structure whose balancing has been deferred;

FIG. 8 is a flow chart illustrating a method of deferred balancing of a tree data structure according to one embodiment of the invention; and

10 FIG. 9 illustrates the tree data structure of FIG. 7 after balancing.

DETAILED DESCRIPTION OF THE INVENTION

[0007] It has been discovered that the execution of operations upon a data structure can be divided into two phases in such manner that the overall efficiency of a computing system is improved. In an update phase, update phase state transitions of operations as further described herein are executed concurrently thus eliminating the need to lock the affected data structure. During the update phase, the data structure can still be traversed using existing links thereby enabling read operations to execute.

[0008] In a commit phase, commit phase state transitions of the operations are executed in an ordered manner. During the commit phase, affected elements of the data structure are locked and the commit phase state transitions are executed atomically. Such transitions include the breaking of existing links thereby precluding the execution of operations requiring navigation of the data structure, such as read operations.

[0009] The system and method of the invention provide for overall improvement in the efficiency of the computing system by reducing the time during which the data structure is locked. Furthermore, the system and method of the invention provide for improved efficiency in the case where the computing system is